LLVMLinux: Embracing the Dragon

Presented by:

Behan Webster

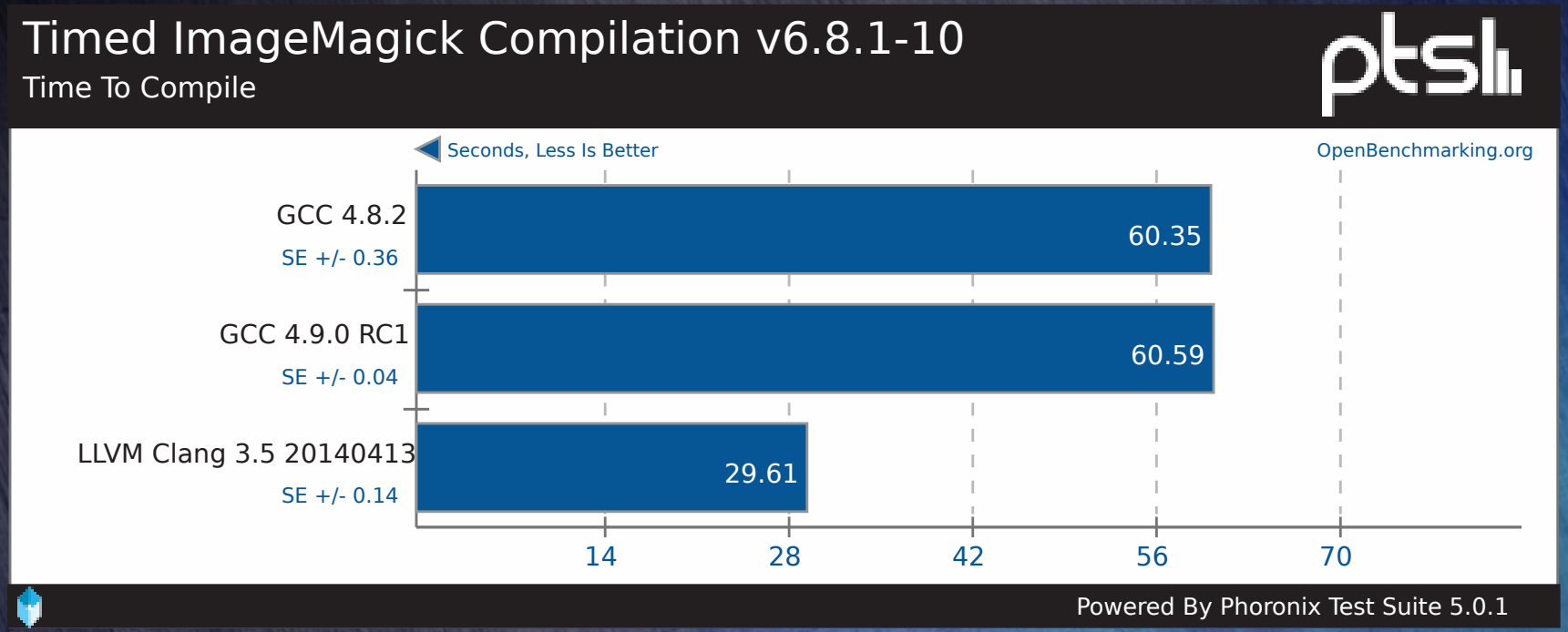
(LLVMLinux project lead)

Presentation Date: 2014.04.30

Clang/LLVM

- LLVM is a Toolchain Toolkit (libraries from which compilers and related technologies can be built)
- Clang is a C/C++ toolchain

New Clang Benchmarks



http://www.phoronix.com/scan.php?page=article&item=gcc49_compiler_llvm35&num=2

Other Interesting LLVM Related Projects

- Official commercial compiler from ARM is based on clang/LLVM
- Clang Static Analyzer
- Energy consumption analysis of programs using LLVM
- llvmpipe (Galium3D)
- CUDA
- OpenCL (most implementations are based on LLVM)
- Clang is one of the Android NDK compilers
- Renderscript in Android is based on LLVM
- Code transformation tools

The LLVMLinux Project Goals

- Fully build the Linux kernel for multiple architectures, using the Clang/LLVM toolchain
- Discover LLVM/Kernel issues early and find fixes quickly across both communities
- Upstream patches to the Linux Kernel and LLVM projects
- Bring together like-minded developers
- Enable the kernel community to do more in depth analysis of the kernel code

LLVMLinux Build/Test System

- Fetches, patches, builds, tests: clang, kernel, qemu, etc
  - git clone http://git.linuxfoundation.org/llvmlinux.git
  - cd llvmlinux/target/vexpress (or x86_64)
  - Make

Patched Mainline Kernel Tree

- A mainline kernel tree with all LLVMLinux patches applied on top is now available:
  - git://git.linuxfoundation.org/llvmlinux/kernel.git
- Dated llvmlinux branches
  - remotes/origin/llvmlinux-2014.03.23
- The master branch is rebased regularly

Linux-Next

- LLVMLinux project now has a branch which is pulled into Linux-Next
  - git://git.linuxfoundation.org/llvmlinux/kernel.git
  - remotes/origin/for-next
- Broader testing of our patches by more people
- Last step before being submitted to mainline

Submission to mainline

- Linux kernel v3.15 had 9 patches pulled directly from the LLVMLinux project
- More patches made it into v3.15 as a part of maintainer trees

Other Avenues of Interest

- Clang Static Analysis of the Linux kernel
- Kernel specific Checkers (GSoC)
- Compiling the Android kernel and AOSP with clang
- Supporting Linaro LLVM team
- Adding clang support to yocto
- Building better tools based on LLVM for the kernel community

LLVMLinux Project Status

- LLVM:
  - One new patch for LLVM (named registers) currently being added by LLVM devs
  - Most LLVM fixes to support the Linux kernel are now being written by upstream maintainers
- Linux Kernel:
  - Roughly 43 kernel patches for various arches

Kernel Patches

- Patches still to upstream (changes since last month)

Architecture	**Number of patches**
all	11 (-6, +1)
arm	12
aarch64	13 (+3)
x86_64	7 (-4, +1)
TOTAL	*43 (-10, +5)*

Aligned Attribute

```
static inline unsigned int shash_align_buffer_size(unsigned len, unsigned long mask)
{
-       return len + (mask & ~(__alignof__(u8 __attribute__ ((aligned))) - 1));
+       typedef u8 __attribute__ ((aligned)) u8_aligned;
+       return len + (mask & ~(__alignof__(u8_aligned) - 1));
}
```

- __attribute__((aligned)) applies the default alignment for the largest scalar type for the target ABI
- gcc allows it to be applied inline to a defined type
- Clang only allows it to be applied to a type definition (PR11071)
- Making it into 2 lines makes it more readable and works in both compilers

Section Mismatch Issues (MergedGlobals)

- By default clang merges globals with internal linkage into one: MergedGlobals
- Allows globals to be addressed using offsets from a base pointer
- Can reduce the number of registers used
- Modpost uses symbol names to look for section mismatches
- MergedGlobals breaks modpost (false positive section mismatches)
- Current solution: use -mno-global-merge to stop global merging
- Updates to modpost may allow this optimization to be enabled again

Section Mismatch Issues (Aliased Symbols)

- Aliased symbols don't inherit __attributes__
- For modules, this means that .init and .exit attributes are dropped when init/exit code is aliased
- Fixed in mainline clang (still an issue in clang v3.4)
- The other option is reapplying __section(.init) and __section(.exit) to aliased symbols

Integrated Assembly Status

- David Woodhouse added .code16 support for X86 ASM
- Renato Golin, Vinicius Tinti, Saleem Abdulrasool and Stepan Dyatkovskiy are working on fixing IA issues in clang to support the Linux ARM kernel code (and ultimately AARCH64)

extern inline: Different for gnu89 and gnu99

- GNU89/GNU90 (used by gcc)
  - Function will be inlined where it is used
  - No function definition is emitted
  - A non-inlined function may also be provided
- GNU99/C99 (used by clang)
  - Function will be inlined where it is used
  - An external function is emitted
  - No other function of the same name may be provided.
- Solution? Use “static inline” instead.

Kbuild

- Basic Kbuild support for clang is in v3.15
- Still need to upstream Kbuild updates for ARM and AARCH64

Named Registers

- Renato Golin has written a patch to add gcc-style named registers to LLVM
- http://llvm-reviews.chandlerc.com/D3261
- This patch will make redundant the 9 named-register patches for ARM and AARCH64
- Adds a new (invalid) warning which still needs fixing
  - warning: variable 'sp' is uninitialized when used here [-Wuninitialized]

Nested Functions

- Patch which removed nested functions from Thinkpad ACPI driver have been Acked by Henrique de Moraes Holschuh
- Patch now in mainline (v3.15)

Variable Length Arrays In Structs

- VLAIS isn't supported by Clang (gcc extension)

```
char vla[n];                    /* Supported, C99/C11 */

struct {
   char flexible_member[]; /* Supported, C99/C11 */
} struct_with_flexible_member;

struct {
   char vlais[n];    /* Explicitly not allowed by C99/C11 */
} variable_length_array_in_struct;
```

- VLAIS is used in the Linux kernel in a number of places, spreading mostly through reusing patterns from data structures found in crypto

Status of VLAIS in the Linux Kernel

- USB Gadget patch is in mainline
- Mac80211 patch is accepted upstream
- Netfilter patch will hopefully be accepted soon
- Patches to remove the use of VLAIS in crypto are now being worked on

Todo Items

- Finish getting Integrated Assembler (IA) working with kernel
- Investigate/fix new clang compiler warnings
- Fix remaining parts that don't yet work:
  http://llvm.linuxfoundation.org/index.php/Broken_kernel_options
- New kernel specific checkers for the clang static analyzer

How Can You Help?

- Make it known you want to be able to use Clang to compile the kernel
- Test LLVMLinux patches
- Report bugs to the mailing list
- Help get LLVMLinux patches upstream
- Work on unsupported features and Bugs
- Submit new targets and arch support
- Patches welcome

Embrace the Dragon. He's cuddly.

Thank you

http://llvm.linuxfoundation.org

Contribute to the LLVMLinux Project

- Project wiki page
  - http://llvm.linuxfoundation.org
- Project Mailing List
  - http://lists.linuxfoundation.org/mailman/listinfo/llvmlinux
  - http://lists.linuxfoundation.org/pipermail/llvmlinux/
- IRC Channel
  - #llvmlinux on OFTC
  - http://buildbot.llvm.linuxfoundation.org/irclogs/OFTC/%23llvmlinux/
- LLVMLinux Community on Google Plus